

pre



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
|-----------------|-------------|----------------------|---------------------|------------------|

09/295,818

04/21/1999

MARTIN HEBERT

1400.9801210

7621

25697

7590

02/03/2004

ROSS D. SNYDER & ASSOCIATES, INC.  
115 WILD BASIN RD.  
SUITE 107  
AUSTIN, TX 78746

EXAMINER

COURTENAY III, ST JOHN

ART UNIT

PAPER NUMBER

2126

/0

DATE MAILED: 02/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

R29

## Office Action Summary

Application No.

09/295,818

Applicant(s)

HEBERT ET AL.

Examiner

St. John Courtenay III

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 21 April 1999.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).  
\* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.  
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

### Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_\_.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other:

### Detailed Action

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1- 37 are rejected under 35 U.S.C. § 102(e) as being clearly anticipated by **Davis et al.** (U.S. Patent 6,225,994).

**Davis** discloses an invention directed to the same problem disclosed in the instant specification. Davis teaches a method for providing reliable interoperation of a plurality of independent software modules, the method comprises the steps of:

- determining selection of an object, by one of the plurality of independent software modules to produce a new object selection, as claimed [e.g., see "determining if the selected object is relevant to it" and associated discussion beginning col. 2, line 15];
- determining object selection status, which indicates a selected object and an associated one of the plurality of independent software modules, as claimed [e.g., see status

information, col. 1, line 41, col. 3, line 59, see node status, col. 14, line 2 and associated discussion]; and

- updating the object selection status when a difference occurs between at least one of the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object, as claimed [e.g., see "reconcile step" and "changing the node status" and associated discussion beginning col. 13, lines 47-53 and con't col. 14, lines 1-19; see also Prototype: BOOL updateForm(struct \*updateInfo), col. 8, line 54; see also "Message to update this object in the ATMC base" col. 10, line 32].
- a user interface (col. 2, line 50, see "GEM", the Graphical Element Manager that provides the GUI component (Graphical User Interface), software modules, processing modules, messages [col. 2, line 1], and selected objects [col. 2, line 15] and elements [col. 2, line 1] operatively coupled as claimed.

-----  
Claims 1- 37 are rejected under 35 U.S.C. § 102(b) as being anticipated by **Montague et al.** (U.S. Patent 5,675,782).

**As per independent claim 1:**

**Montague** teaches a method for providing reliable interoperation of a plurality of independent software modules, the method comprises the steps of:

- a) determining selection of an object by one of the plurality of independent software modules to produce a new object selection [e.g., see "objects" and "access rights" and

associated discussion beginning col. 10, line 10, see changing the access rights by a user, col. 10, line 35];

- b) determining object selection status, which indicates a selected object and an associated one of the plurality of independent software modules [e.g., see changing the access rights by a user, col. 10, line 35 and associated discussion]; and
- c) updating the object selection status when a difference occurs between at least one of the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object [e.g., see "granting access rights to an object or container for one or more trustees" and associated discussion beginning col. 11, line 19].

**As per independent claim 12:**

**Montague** teaches a method for providing reliable interoperation of a plurality of independent software modules, the method comprises the steps of.

- a) maintaining object selection status that includes identity of a selected object and identity of an associated one of the plurality of independent software modules, wherein the associated one of the plurality of independent software modules selected the selected object [e.g., see "objects" and "access rights" and associated discussion beginning col. 10, line 10, see changing the access rights by a user, col. 10, line 35];
- b) detecting selection of a function by an independent software module of the plurality of independent software

modules [e.g., see changing the access rights by a user, col. 10, line 35 and associated discussion]; and

- c) performing, by one of the plurality of independent software modules, the function upon the selected object [e.g., see "granting access rights to an object or container for one or more trustees" and associated discussion beginning col. 11, line 19].

**As per independent claim 20:**

**Montague** teaches a software management system that comprises:

- a plurality of independent software modules and a processing module [see "plurality of APIs", col. 9, line 67]; and,
- memory [inherent in the networked computer system disclosed by Montague, col. 4] operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to:
  - (a) determine selection of an object by one of the plurality of independent software modules to produce a new object selection [e.g., see "objects" and "access rights" and associated discussion beginning col. 10, line 10, see changing the access rights by a user, col. 10, line 35];
  - (b) determine object selection status, which indicates a selected object and an associated one of the plurality of independent software modules [e.g., see changing the access rights by a user, col. 10, line 35 and associated discussion]; and

- (c) update the object selection status when a difference occurs between at least one of the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object [e.g., see "granting access rights to an object or container for one or more trustees" and associated discussion beginning col. 11, line 19].

**As per independent claim 30:**

**Montague** teaches a software management system comprises:

- a plurality of independent software modules and a processing module [see "plurality of APIs", col. 9, line 67];
- memory [inherent in the networked computer system disclosed by Montague, col. 4] operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to:
  - (a) maintain object selection status that includes identity of a selected object and identity of an associated one of the plurality of independent software modules, wherein the associated one of the plurality of independent software modules selected the selected object [e.g., see "objects" and "access rights" and associated discussion beginning col. 10, line 10, see changing the access rights by a user, col. 10, line 35];
  - (b) detect selection of a function by an independent software module of the plurality of independent software modules [e.g., see changing the access rights by a user, col. 10, line 35 and associated discussion]; and

- (c) performing, by one of the plurality of independent software modules, the function upon the selected object [e.g., see "granting access rights to an object or container for one or more trustees" and associated discussion beginning col. 11, line 19].

**As per dependent claim 2:**  
**Montague** teaches:

- evoking a function by an independent software module of the plurality of independent software modules [e.g., see "GrantAllAccessRights" function and associated discussion col. 11, line 20] ; and
- performing, by the independent software module, the function upon the selected object identified in the object selection status [see setting the object access rights, col. 11, line 33].

**As per dependent claim 3:**

**Montague** teaches obtaining, by the independent software module, the function from the associated one of the plurality of independent software modules [e.g., see "ReplaceAllAccessRights" function and associated discussion col. 11, line 31].

**As per dependent claim 4:**

**Montague** teaches informing, by the independent software module, the associated one of the plurality of independent software modules to perform at least one of: the function upon the selected object and a corresponding function upon the selected object [e.g., see "IsAccessPermitted" function and associated discussion col. 11, line 49].



**As per dependent claim 5:**

**Montague** teaches updating the object selection status with at least one of the new object selection and the one of the plurality of independent software modules [see discussion col. 11, lines 18-29].

**As per dependent claim 6:**

**Montague** teaches providing a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module [e.g., see "RevokeExplicitAccessRights" function and associated discussion col. 11, line 36].

**As per dependent claim 7:**

**Montague** teaches receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling software module [e.g., see "EnumerateTrustees" function and associated discussion col. 12, line 12].

**As per dependent claim 8:**

**Montague** teaches providing a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module [e.g., see "GetEffectiveAccessRights" function and associated discussion col. 11, line 67].

**As per dependent claim 9:**

**Montague** teaches the plurality of independent software modules includes at least one network management software module and a plurality of element network software modules, wherein the at least one network software module performs network level

operations for a communication network and wherein each of the plurality of element network software modules performs area level operations for a corresponding portion of the communication network [e.g., see "network configuration" and associated discussion col. 5, lines 13-51] .

**As per dependent claim 10:**

**Montague** teaches the plurality of independent software modules includes computer applications, wherein at least some of the computer applications control independent user interfaces [e.g., see "dialog boxes" and associated discussion entire col. 9, beginning line 1].

**As per dependent claim 11:**

**Montague** teaches the object includes at least one object element [see object discussion beginning line 11, col. 10].

**As per dependent claim 13:**

**Montague** teaches step further comprises detecting selection via a user interface associated with the independent software module [col. 9, lines 43-51].

**As per dependent claim 14:**

**Montague** teaches receiving notification of the function from another one of the plurality of independent software modules, wherein the another one of the plurality of independent software modules detected selection of the function via a user interface associated with the one of the plurality of independent software modules [e.g., see user selection of object access rights and associated discussion col. 10, lines 35 - 67].

**As per dependent claim 15:**

**Montague** teaches:

- determining selection of a new selected object by one of the plurality of independent software modules to produce a new object selection [e.g., see changing the access rights by a user, col. 10, line 35 and associated discussion];
- determining current object selection status [e.g., see "GetEffectiveAccessRights" function and associated discussion col. 11, line 67]; and
- updating the object selection status when a difference occurs between at least one of the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object [see discussion col. 11, lines 18-29].

**As per dependent claim 16:**

**Montague** teaches updating the object selection status with at least one of the new selected object and the one of the plurality of independent software modules [see discussion col. 11, lines 18-29].

**As per dependent claim 17:**

**Montague** teaches providing a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module [e.g., see "RevokeExplicitAccessRights" function and associated discussion col. 11, line 36].

**As per dependent claim 18:**

**Montague** teaches receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling

software module [e.g., see user selection of object access rights and associated discussion col. 10, lines 35 - 67].

**As per dependent claim 19:**

**Montague** teaches providing a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module [e.g., see "ReplaceAllAccessRights" function and associated discussion col. 11, line 31].

**As per dependent claim 21:**

See the rejection of claim 2 detailed above.

**As per dependent claim 22:**

See the rejection of claim 3 detailed above.

**As per dependent claim 23:**

See the rejection of claim 4 detailed above.

**As per dependent claim 24:**

See the rejection of claim 5 detailed above.

**As per dependent claim 25:**

See the rejection of claim 6 detailed above.

**As per dependent claim 26:**

See the rejection of claim 7 detailed above.

**As per dependent claim 27:**

See the rejection of claim 8 detailed above.

**As per dependent claim 28:**

See the rejection of claim 9 detailed above.

Application/Control Number:  
09/295,818  
Art Unit: 2126

Page 12

**As per dependent claim 29:**

See the rejection of claim 11 detailed above.

**As per dependent claim 31:**

See the rejection of claim 13 detailed above.

**As per dependent claim 32:**

See the rejection of claim 14 detailed above.

**As per dependent claim 33:**

See the rejection of claim 15 detailed above.

**As per dependent claim 34:**

See the rejection of claim 5 detailed above.

**As per dependent claim 35:**

See the rejection of claim 6 detailed above.

**As per dependent claims 36 and 37:**

See the rejection of claim 7 detailed above.

Application/Control Number:  
09/295,818  
Art Unit: 2126

Page 13

**Prior Art not relied upon:**

Please refer to the references listed on the attached PTO-892 which are not relied upon in the claim rejections detailed above.

Application/Control Number:  
09/295,818  
Art Unit: 2126

Page 14

### **How to Contact the Examiner:**

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to **St. John Courtenay III** whose voice telephone number is **(703) 308-5217**. A voice mail service is also available at this number. Normal Flex work schedule: M – F 7:30 AM - 4:00 PM

- **All responses sent by U.S. Mail should be mailed to:**

Commissioner for Patents  
PO Box 1450  
Alexandria, VA 22313-1450

---

### **Patent Customers advised to FAX communications to the USPTO**

<http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/faxnotice.pdf>

**Effective Oct. 15, 2003, ALL patent application correspondence transmitted by FAX must be directed to the new PTO central FAX number:**

**NEW PTO CENTRAL FAX NUMBER:  
703-872-9306**

- 
- Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: (703) 305-3900.**

**Please direct inquiries regarding fees, paper matching, and other issues not involving the Examiner to:**

**Technical Center 2100 CUSTOMER SERVICE: 703 306-5631**

The Manual of Patent Examining Procedure (MPEP) is available online at:  
<http://www.uspto.gov/web/offices/pac/mpep/index.html>

  
ST. JOHN COURTENAY III  
PRIMARY EXAMINER